

MULTI-PHASE SAMPLING

BACKGROUND OF THE INVENTION

5 The present invention relates to multi-phase sampling. In particular, the present invention provides a multi-phase sampling system that overcomes problems with current multi-phase systems that result from the inability of these multi-phase sampling systems to optimally sample an incoming data signal.

10 An ever-increasing demand exists for communications systems that are capable of operating at increasingly higher data rates. As monolithic processes (e.g., complementary metal oxide semiconductor (CMOS)) are increasingly being used to create devices that perform high-speed data processing, it has become necessary to use multiple phases in the devices for sampling and processing incoming signals in order to sample the incoming data signals at the Nyquist rate. The need for using
15 multiple samplers in order to sample the data signal at a high enough speed can be seen from the following example. The cycle or retrigger time of a data retiming latch is often limited by the regeneration speed of the internal positive feedback circuitry of the latch. Although such a sampler or latch cannot be retriggered fast enough to Nyquist sample an incoming signal of a particularly high data rate, it is capable of
20 taking a snapshot of a rapidly changing signal. By using a multiplicity of samplers on evenly staggered clock phases, each latch can be allowed a generous regeneration time, while still enabling sampling of the data signal at the Nyquist rate. An example of a known multi-phase system is a 3.5 gigabit per second (Gb/s) retiming circuit for non-return-to-zero (NRZ) data that uses a 10-phase sampling system built in 0.28
25 micrometer (um) CMOS. Such a system is disclosed in ISSCC Digest of Technical Papers, Vol. 42, pages 352-353 and 478, February 15-17, 1999.

30 These and similar types of systems share the strategy of processing high-speed incoming signals by using multiple lower-speed samplers that all sample the incoming signal in a round-robin fashion. The samplers are lower in speed because they are comprised of larger transistors, which have larger parasitic capacitances, and thus have longer regeneration times and slower retrigger times. Due to the slower speeds of the large samplers, many samplers may be required in order to sample a very high data rate signal at the Nyquist rate. Also, because the samplers are larger in size, they

dissipate more power. Obviously, such systems have many disadvantages that need to be overcome.

Multi-phase systems are also used to transmit data. In these systems, data is transmitted by using multiple phases to control respective selectors, each of which gates a different bit to the output of the transmitting multi-phase system. In this case, the jitter of each data edge is determined by the timing error of each data phase.

For monolithic implementations (e.g., CMOS), it is common to generate the multiple phases using a ring oscillator. The most common form of ring oscillator utilizes a number N of identical gain stages, each with a delay of τ , connected in a ring with a net inversion. As is well known in the art, such a system oscillates with a period of $2N\tau$, and produces $2N$ evenly-spaced timing phases, with the phases being derived from both the rising and falling edges of each delay cell output. Such oscillators can be made tunable and are well-suited for monolithic IC implementation. However, such systems are designed such that the number of phases is even, i.e., for $2N$ evenly-spaced phases, where N is an integer.

In practice, the exact delay of oscillator elements will differ between the N different cells. These variations include random cycle-to-cycle delay variations, resulting largely from thermal, supply, and substrate noise in the devices, and deterministic delay errors, which are consistent cycle after cycle. These static delay errors can be caused by a number of factors, including: 1) device size variation due to fabrication errors that occur during the IC fabrication process, 2) non-symmetrical layout of the various delay cells, including capacitance, resistance and inductive mismatches caused by unequal wiring or crosstalk with other signals, 3) unequal proximity of various delay cells to other features which causes variations in doping or dielectric thickness to occur and/or unequal loading of the various delay outputs which can cause mismatched fan-out delays.

In addition, these same delay variations can afflict the clock distribution network. Because of its distributed nature, the clock network may also suffer from delay mismatches due to unequal supply voltages caused by resistive drops across the power distribution network. Also, it is possible for variations in the clock-to-sample delay between the input samplers to occur. The sum of all these errors results in an overall phase sampling error in the system.

One known way of preventing or reducing these phase errors is to decrease the
aforementioned error factors 1) – 3). However, solutions for doing this are difficult to
implement and, thus far, have not been totally satisfactory. Furthermore,
implementing these solutions typically result in increases in power dissipation and
5 circuit size, and/or place an onerous burden on the designer to create precisely
symmetrical samplers. Furthermore, such solutions must be implemented during
device design and fabrication, not after the device has already been created.
Therefore, failure to satisfactorily implement the solutions during the design and
manufacturing process will likely result in phase error sampling problems occurring
10 during operation of the device.

Another approach that has been used to eliminate phase error sampling
problems in multi-phase systems is to focus on ensuring that the clock is very precise
and to use very large samplers (referred to herein interchangeably as “latches”). If
relatively small latches are used with a very precise clock, sampling errors will still
15 occur due to the fact that the parasitic capacitances of the smaller transistors of the
smaller latches result in very significant timing mismatches between the latches.
However, if very large latches are used, which means that very large transistors are
used to create the latches, the latches are slower and the timing mismatches caused by
the differences in the parasitic capacitances of the transistors of the latches are less
20 significant. Therefore, using very large latches with a very precise clock can decrease
or eliminate sampling problems caused by phase errors, but there is a tradeoff. As
stated above, larger latches take up more area, are slower than smaller latches and also
consume more power than smaller latches, which are all undesirable traits in circuit
design.

Accordingly, a need exists for a multi-phase system that is capable of
sampling high speed signals at the Nyquist rate and that does not require the use of
large samplers, which normally have relatively high power consumption
requirements. A need also exists for a multi-phase system that does not require that
the clock be precise or that the layout for each sampler be completely symmetrical,
30 thus eliminating onerous burdens that would otherwise be placed on designers and
manufacturers to create perfectly symmetrical latches and/or perfectly precise clocks.

SUMMARY OF THE INVENTION

In accordance with the present invention, a multi-phase sampling system is provided that utilizes samplers that each sample both a transition and data of the incoming data signal. By allowing each sampler to sample a transition and data of the incoming data signal the multi-phase sampling system is capable of optimizing the sampling tasks performed by the samplers.

In accordance with one embodiment of the present invention, an odd number, n , of samplers are used to sample an incoming data signal. A multi-phase clock generator generates n clock signals of n evenly staggered respective phases. Phase error determination circuitry may then be used to obtain phase errors for each sampler.

If the objective is to correct the phase errors of the apparatus sampling the incoming data signal, such as a receiver, for example, the phase errors are utilized by phase shifting circuitry of the apparatus in a feedback loop to drive the phase errors to zero. In the case where the phase error determination circuitry of the apparatus includes circuitry for determining the phase errors associated with a multi-phase system that is transmitting the data signal, i.e., a transmitter, the phase error determinations made by the apparatus for the transmitter may be sent by the apparatus back to the transmitter to enable the multi-phase transmitter to correct its own phase errors, thereby improving the integrity of the transmitted signal.

This embodiment assumes that the multi-phase transmitter incorporates phase shifting circuitry and feedback loop similar to that incorporated into the multi-phase receiver to enable the multi-phase transmitter to perform the necessary phase error corrections once it receives the phase error determinations. This also assumes that the multi-phase system sampling the received signal have some way of knowing the relative phases of the transmitting multi-phase system, which can be accomplished in a multiplicity of ways, as discussed below in more detail. An alternative to providing the relative phases of the multi-phase transmitter to the multi-phase receiver would be to provide the transmitting multi-phase system with its own phase error determination circuitry in addition to its own phase shifting and feedback circuitry.

In accordance with one embodiment, the apparatus of the present invention is a multi-phase system that comprises at least first, second and third sampling devices, phase error determination circuitry for determining the phase errors associated with each of the samplers of the apparatus and phase shifting circuitry for shifting one or

more of the phases in order to drive any phase errors to zero. The associated method includes the steps of sampling a first data signal with first, second and third sampling devices upon receiving first, second and third clock signals of first, second and third phases, respectively, and outputting first, second and third output signals,

5 respectively. Once these steps have been performed, first, second and third phase error determinations associated with the first, second and third output signals are then determined. Then, at least one of the first, second or third phases is shifted in accordance with the respective first, second or third phase error determinations to eliminate or reduce phase errors in the multi-phase system.

10 In accordance with another embodiment of the present invention, the method and apparatus are implemented in conjunction with a receiver and/or a transmitter and/or a transceiver. In these cases, the phase error determinations of the multi-phase receiver are binned, modulo the number of phases of the multi-phase receiver. Likewise, the phase error determinations of the multi-phase transmitter are binned,
15 modulo the number of phases of the multi-phase transmitter. The phase error determinations corresponding to the receiver are then used by the receiver to drive its own phase errors to zero and the phase error determinations of the transmitter are fed back to the transmitter (or transmitter portion of the transceiver) to inform the transmitter as to how much to adjust its clock phases in order to drive its own phase
20 errors to zero.

In all cases, preferably every sampler samples a transition and data, which enables phase error determinations to be made for every sampler. By allowing phase error determinations to be made for every sampler, the phase error determinations can be used during calibration and in real-time operations to drive the phase errors of the
25 multi-phase system to zero. Therefore, problems that can result from phase errors do not need to be addressed during the design and fabrication stages. Furthermore, the present invention also eliminates the need to use large samplers, which normally have relatively high power consumption requirements, in order to avoid problems associated with phase errors. Also, problems associated with trying to ensure that the
30 multi-phase system clock is extremely precise and/or that the layout for each sampler is completely symmetrical are obviated.

These and other features and embodiments of the present invention will be described below with reference to the detailed description, drawings and claims.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 illustrates a block diagram and associated timing chart for a multi-phase system that uses an odd number of latches to sample a high-speed data stream at the middle and transition points of each bit.

5 Fig. 2 illustrates a block diagram of the multi-phase clock signal generation circuit of the present invention.

Figs. 3A - 3D are block diagrams illustrating various uses of the present invention in transmitters and/or receivers and/or transceivers.

DETAILED DESCRIPTION OF THE INVENTION

10 The apparatus of the present invention can be implemented by a fully-monolithic technique. Thus, the apparatus and method of the present invention can be implemented in one or more ICs that enable systematic phase errors in a multi-phase system to be reduced or eliminated via precise calibration. However, as will be
15 understood by those skilled in the art from the following description, the present invention applies also to non-monolithic applications and, furthermore, is applicable to any system or application where multiple samplers are needed to sample a signal.

When sampling an analog data voltage signal with a latch (also referred to herein as a sampler) to convert the analog data voltage signal into a digital data
20 voltage signal, there is a precise time at which the latch should sample the signal. The sampling of the analog data voltage signal by the latch is triggered by a clock signal generated by an oscillator. Ideally, the clock signal should arrive at the latch at the precise time that the latch should sample the analog data voltage signal. However, in reality, it is unlikely that the clock signal will arrive at the precise time that the analog
25 data voltage signal should be sampled. This is because, in a multi-phase system, when clock phases are created and routed to latches, errors inherent in the physical structure of the latches and in the oscillator often cause the latches not to be triggered at the precise time necessary to cause the latches to optimally sample the data signal.

Fig. 1 is an example of a multi-phase system in accordance with the present
30 invention that uses three latches (also referred to herein interchangeably as “samplers”) to sample a high-speed data stream at both the middle (Dk) and transition (Tk) of each bit. This differs from known techniques, which employ an even number of samplers. The example system of Fig. 1 utilizes three 2 GHz clock phases, collectively designated by the numeral 1 for sampling an incoming 3 Gb/s data signal

designated by the numeral 2, at both the middle of the bit and at the transitions between bits, as shown in the timing diagram 3 of Fig. 1. Each of the three latches 4, 5 and 6 receives one of the three phase-shifted clock signals 1. The timing diagram 3 and the outputs of the latches 4, 5 and 6 indicate the manner in which the high speed input signal 2 is capable of being sampled at the Nyquist rate.

The manner in which the multi-phase system of Fig. 1 operates can be understood by looking at the outputs 7, 8 and 9 of the latches and 4, 5 and 6 at the timing diagram 3. Latch 4 samples at clock phase ϕ_0 . Latch 5 samples at clock phase ϕ_1 . Latch 6 samples at clock phase ϕ_2 . The timing at which each latch samples the middle of a bit or a bit transition is shown in the timing diagram 3. The outputs 7 of latch 4, which samples at clock phase ϕ_0 , are mid-bit sample D0, transition bit sample T2, mid-bit sample D3 and transition-bit sample T5, etc. The outputs 8 of latch 5, which samples at clock phase ϕ_1 , are transition-bit sample T1, mid-bit sample D2, transition bit sample T4, and mid-bit sample D5, etc. The outputs 9 of latch 6, which samples at clock phase ϕ_2 , are mid-bit sample D1, transition bit sample T3, mid-bit sample D4 and transition-bit sample T6, etc. Thus, it can be seen that this multi-phase system samples the data at the Nyquist rate of 6 GHz (i.e., twice the data rate of 3 GHz), and that each sampler samples a transition. By causing each sampler to sample a transition, phase errors can be determined for each of the samplers, as discussed below in more detail.

In accordance with the present invention, it has been determined that the timing of the clock phase signals can be adjusted in accordance with measured sampling timing errors to cause each of the clock phase signals to arrive at the latches at precisely the correct points in time that the latches should be triggered in order to optimally sample the analog data voltage signal and convert it into a digital voltage signal.

Some of the errors for which the multi-phase system should adjust can be mathematically described as follows. Assuming an oscillator "A" having a fundamental oscillation frequency f_A , and generating m distinct phases labelled:

$$\Phi_i^A; i \in \{0, 1, 2, \dots, (m-1)\},$$

where each i^{th} phase exhibits a static deterministic phase error $\epsilon_i^A; i \in \{0, 1, 2, \dots, (m-1)\}$ with respect to its ideal phase and subject to the constraint that

$\sum \varepsilon_i^A = 0$ due to various circuit mismatches. In addition, because of thermal noise and other random phenomenon, each timing event has an associated additive zero-mean random phase error, which is not correlated between timing events. The present invention eliminates or reduces the deterministic portion of these errors.

5 The multi-phase clock generation circuit of the present invention, in accordance with an example embodiment, will now be described with reference to Table 1 below and with reference to the block diagram of Fig. 2. Table 1 is a truth table that indicates that the logical outputs of multiple latches (e.g., three latches) obtained at a given point in time can be used to determine whether a timing error has
10 occurred. These errors are referred to as “early-late” errors. This truth table is known in the art as an Alexander Phase Determination Truth Table. The early-late error determination technique takes two adjacent bit samples along with an intermediate bit transition sample and logically combines these samples in a known manner to determine whether the phase of the transition is either early or late.

15

TABLE 1: Truth table for random data early-late ATB phase detector

Mid Bit Sample (A)	Transition Sample (T)	Mid Bit Sample (B)	Phase Error Determination (ATB Result)
0	0	0	no transition
0	0	1	Early
0	1	0	Invalid
0	1	1	Late
1	0	0	Late
1	0	1	Invalid
1	1	0	Early
1	1	1	no transition

The early-late error results obtained from logically combining the samples are indicated in the last column of Table 1. However, these results provide no guidance as to how much the phase offsets should be adjusted in order to eliminate the errors.
20 In accordance with the present invention, the early-late error determinations are used in a feedback loop to adjust the clock offsets in order to eliminate the errors so that the timing at which the clock signals trigger the latches, or samplers, is correct and

precise. In essence, indications of the early-late phase errors are binned into separate charge pumps, modulo the number of phases of the multi-phase system, to enable the clock offsets to be properly and precisely adjusted.

As shown in the example system 10 represented by the block diagram of Fig. 2, three latches 11, 12 and 13 receive the same data signal 14. The latches 11, 12 and 13 are triggered by clock signals 14, 15 and 16, respectively, at clock signal inputs 17, 18 and 19, respectively. The multiphase clock signals 14, 15 and 16 (ϕ_0 , ϕ_1 and ϕ_2) are generated by a three-phase oscillator 20. It should be noted that although the system shown in Fig. 2 is a three-phase system, the present invention is not limited to any particular number of phases, or to odd or even phases. The three-phase system 10 of Fig. 2 is being used merely to provide an example of the manner in which the present invention can be implemented.

An ATB detector 30 comprises logic that performs the functions described above with reference to Table 1 to determine whether an early-late error has occurred. The term "ATB" is a short-hand acronym that is generally used in the art to describe the process represented by Table 1, in which the left column is referred to as "A" for the mid-bit A sample, the middle column is referred to as "T" for the transition sample of bit A, and the right column is referred to as "B" for the mid-bit B sample. The ATB detector 30 receives the digital outputs 21, 22 and 23 from latches 11, 12 and 13, respectively, and generates outputs 24, 25 and 26. Outputs 24, 25 and 26 correspond, respectively, to ϕ_0 , ϕ_1 and ϕ_2 error samples. In the example shown in Fig. 2, the ϕ_2 error samples are used to phase-lock the system 10 to the data signal, and the ϕ_0 and ϕ_1 error samples are then adjusted via phase shifters 31 and 32, respectively, to be in alignment with ϕ_2 . The phase shifters 31 and 32 receive the outputs from charge pump integrators 33 and 34, respectively, which bin the errors for phase shifts ϕ_0 and ϕ_1 , respectively. The charge pump integrators 33 and 34 have associated capacitors, as pictorially indicated, and are configured to low pass filter the early-late phase errors and average them (i.e., integrate them) over time.

The charge pump integrators 33 and 34 receive at their inputs the early-late errors in the form of binary signals. If the phase error binary signal received at the input of a charge pump integrator corresponds to an early error indication, a fixed amount of charge is placed on the capacitor associated therewith. If the phase error binary signal received at the input of a charge pump integrator corresponds to a late

error indication, a fixed amount of charge is removed from the capacitor associated therewith.

Generally, the example circuit 10 shown in Fig. 2 is calibrated in accordance with the ϕ_2 early-late errors by passing those error signals through a phase-locked
 5 loop (PLL) filter 35, which provides feedback to the oscillator 20. The oscillator 20 then generates the three phase clock signals based on the timing of the feedback signal received thereby while maintaining the alignment of the three phase clock signals. Once the circuit 10 has been phase-locked to the ϕ_2 early-late errors, then the phase
 10 adjusters 31 and 32 only need to make small phase shifts to ϕ_0 and ϕ_1 to ensure that all three latches are triggered at precisely the correct time. Ideally, the feedback loop in Fig. 2 causes the phase errors to be driven to zero.

It should also be noted that it is not necessary that the apparatus of the present invention can be implemented as a phase-locked system. The apparatus could instead be implemented as a delay-locked system, in which case, for a three-phase system, for
 15 example, in which the frequency is known a priori, three phase shifters would be used, rather than two, as shown in the example of Fig. 2. In the delay-locked system, delay elements would be used to determine the phases and then each of the phases would be adjusted by the respective phase shifters, if necessary, in accordance with the respective early-late phase errors. Alternatively, if the frequency is not known,
 20 then, in the case of a three-phase system, for example, the sum of all of the early-late error indications would be used to determine the frequency and lock onto it. Subsequently, a three-phase shifter would adjust each of the respective phases in accordance with the early-late error indications. Those skilled in the art will understand, in view of the discussion provided herein, the manner in which such
 25 alternate forms of the invention could be implemented.

It can be seen from the example described above with reference to Fig. 2 that separate charge pump integrators are used to bin the error indications, and that the number of charge pumps utilized is based on the modulus (i.e., the number of phases) of the multi-phase system. This binning process is one of the primary features of the
 30 present invention because it is what makes precise adjustment of clock offsets possible so that the sampling of the latches (or, in other words, samplers) is precisely performed. This binning process is also referred to herein as modulo binning.

In accordance with another embodiment of the present invention, the method and apparatus are implemented in conjunction with a receiver and/or a transmitter

and/or a transceiver. These variations of the present invention will be discussed with reference to Figs. 3A – 3D. In these cases, the phase error indications of the multi-phase system of the receiver can be binned into separate charge pumps, modulo the number of phases of the receiver multi-phase system, in order to enable the receiver clock offsets to be precisely adjusted. Likewise, the phase error indications of the multi-phase system of the transmitter can be binned into separate charge pumps, modulo the number of phases of the transmitter multi-phase system, in order to enable the transmitter clock offsets to be precisely adjusted. The location at which these modulo binning processes are performed is not limited to any particular location. For example, as discussed below in more detail, the receiver can have circuitry such as that shown in Fig. 2 that obtains the phase error indications corresponding to the transmitter multi-phase system and that provides feedback to the transmitter to inform the transmitter as to how much to adjust its clock offsets.

In order to demonstrate the manner in which the present invention can be utilized in these receiver/transmitter/transceiver embodiments, an example will now be described with reference to Tables 2 and 3 in which edges from one n -phase oscillator are used to sample the edges of another m -phase oscillator, where m and n are relatively prime. The phase errors from this cross-sampling are then processed to derive specific phase errors for each of the $m + n$ phases of both the transmitter and receiver oscillators. Once the phase errors have been derived, the binning process of the present invention discussed above with reference to Fig. 2 is used to drive the measured phase errors to zero for all the phases of both oscillators.

For purposes of describing this embodiment, it will be assumed that the receiver (referred to herein as “receiver” or “RX”) samples the transmitter (referred to herein as “transmitter” or “TX”) data at twice the transmitter data bit rate, such that even numbered samples are aligned to the transitions of the TX waveform and odd numbered samples are aligned to the mid-bit of the TX waveform. In accordance with this example embodiment of the present invention, two oscillators, one with m phases, and one with n phases (where m , n are relatively prime) are used to cross sample each other’s phase errors. The present invention allows a set of phase errors to be decomposed into errors specific to each of the m and n phases by using the ATB logic and modulo binning process described above with reference to Fig. 2. Feedback loops such as that described above with reference to Fig. 2 are then used to

independently drive the $m + n$ error indications to zero by making adjustments to phase shifters in each of the $m + n$ outputs.

An example of this embodiment would be a system comprising a transceiver for serial data communication that comprises an m phase oscillator used for the transmitter clock and the n phase oscillator used for the receiver clock. By using the receiver to sample the transmitter output, the phase correction signals can be generated with minimal additional circuitry. In systems such as a multi-phase analog-to-digital converter (ADC), the secondary oscillator is used solely for calibration purposes. In systems where the transmitter phase is not controllable, the receiver preferably is built with an odd number of phases. This is because multi-phase transmitter systems almost universally use 2^k phases, $k \in \{1, 2, 3, \dots\}$, and using an odd number of receiver phases improves the probability that m, n will be relatively prime. A simple case of such a system is shown in Fig. 3A, in which a stand-alone receiver calibrates itself without any communication to the associated transmitter. This would typically be the case of a receiver that is required to be interoperable with industry-standard transmitter parts which may not be capable of calibration, or in a simplex application where there is no means of communication from the receiver, RX, to the transmitter, TX.

If the RX has an oscillator with an odd number of phases, then all phases will eventually be used to sample edges of the incoming data sent from the TX. In this case, odd receiver timing events are used to sample transmitter transitions, which are determined by the transmitter timing events. These phase error indications are preferably low pass filtered in the manner discussed above with reference to Fig. 2, and used in a PLL such as that shown in Fig. 2 to force the mean error between TX edges and RX samples to zero. If m, n are relatively prime, then the deterministic portion of the phase error E_k is periodic with period $m*n$ phase samples, and the average error can be computed by summing over $m*n$ samples by the following equation:

$$\bar{E} = \frac{1}{mn} \sum_{i=0}^{mn-1} E_i = \frac{1}{mn} \sum_{i=0}^{mn-1} (\varepsilon_{i \bmod n}^B - \varepsilon_{i \bmod m}^A) \quad \text{Equation 1}$$

This can be more intuitively understood by enumerating the phase errors in a simple example. If $m = 4$ and $n = 3$, then the deterministic phase errors will repeat in a cycle of $4 \times 3 = 12$. Table 2 shows the progression.

TABLE 2.

I	TX phase	RX phase
0	0	0
1	1	1
2	2	2
3	3	0
4	0	1
5	1	2
6	2	0
7	3	1
8	0	2
9	1	0
10	2	1
11	3	2

5

By inspection, the rightmost term in Equation 1 can be refactored either as

$$\bar{E} = \frac{1}{mn} \sum_{i=0}^{n-1} \left(m \varepsilon_i^B - \sum_{j=0}^{m-1} \varepsilon_j^A \right) \quad \text{Equation 2}$$

corresponding to Table 3 below, or as:

$$\bar{E} = \frac{-1}{mn} \sum_{j=0}^{m-1} \left(n \varepsilon_j^A - \sum_{i=0}^{n-1} \varepsilon_i^B \right) \quad \text{Equation 3}$$

10 corresponding to Table 4 set forth below Table 3.

15

TABLE 3.

TX phase	RX phase	Interpretation
0	0	The average of these states provides an estimate for error in RX phase 0
1	0	
2	0	
3	0	
0	1	The average of these states provides an estimate for error in RX phase 1
1	1	
2	1	
3	1	
0	2	The average of these states provides an estimate for error in RX phase 2
1	2	
2	2	
3	2	

TABLE 4.

TX phase	RX phase	Interpretation
0	0	The average of these states provides an estimate for error in TX phase 0
0	1	
0	2	
1	0	The average of these states provides an estimate for error in TX phase 1
1	1	
1	2	
2	0	The average of these states provides an estimate for error in TX phase 2
2	1	
2	2	
3	0	The average of these states provides an estimate for error in TX phase 3
3	1	
3	2	

The second summation in the parenthesized portions of Equations 2 and 3 are the sum of all the TX and RX phase error terms, respectively. Because the loops are frequency and phased locked, these phase error terms are zero mean and sum to zero over any block of $m*n$ phase samples. Thus, each of the terms of the first summation are equal to each of the n phase errors of the RX and to the m phase errors of the TX. These terms can be easily applied to the phase shifters in the feedback loop to drive the measured phase errors to zero, as will be understood by those skilled in the art in view of the discussion provided herein. Therefore, in accordance with the present invention, the error samples are thereby decimated by either m or n and averaged to produce $m + n$ separate phase error estimates. Each of these estimates will have a mean value proportional to the phase error of each of the TX and RX oscillator phases, respectively.

A short software simulation algorithm that simulates the method of the present invention and that has been coded in the interpreted language AWK (a language named for the initials of its authors) will now be described. The code begins with defining the number of phases for the RX and TX, setting an adaptation coefficient for the feedback loop, setting the number of time steps to be simulated, and initializing a random number generator to generate some number of phase errors:

```

M=5                # number of phases for RX
N=4                # number of phases for TX
EPS=0.003          # adaptation coefficient
MAXTIME=2000       # number of steps to simulate
srand();

```

An array is set up to hold the M random phase errors for the RX oscillator, and is initialized to random starting phases. A mean phase error is computed and a correction is subtracted from all the phases to force the mean to zero:

```

# create random phase errors for RX osc
total=0.0
for (i=0; i<M; i++) {
    a[i]=rand()-0.5
    total+=a[i]
}
for (i=0; i<M; i++) { # force zero mean

```

```

    a[i] -= total/M
}

```

The same procedure is followed to initialize the N phase errors for
 5 the TX oscillator:

```

# create random phase errors for TX osc
total=0.0
10 for (i=0; i<N; i++) {
    b[i]=rand()-0.5
    total+=b[i]
}
for (i=0; i<N; i++) { # force zero mean
15     b[i] -= total/N
}

```

In the language AWK, the modulo function is indicated by “%”, and so “k mod m” becomes “k%m”. The main adaptation loop is then rendered as:

```

20 for (i=0; i<MAXTIME; i++) {
    aa[i%M] += EPS*sign((a[i%M]-aa[i%M]) -
        (b[i%N]-bb[i%N]))
    bb[i%N] += EPS*sign((b[i%N]-bb[i%N]) -
25     (a[i%M]-aa[i%M]))
    print i, a[i%M]-aa[i%M], "a" i%M
    print i, b[i%N]-bb[i%N], "b" i%N
}

```

30 The values stored in the arrays “aa[i%M]” and “bb[i%N]” are updated in the code steps of EPS and are used to develop estimates of the phase error for each oscillator phase. In practice, these values would be used to drive phase shifter circuits (e.g., elements 31 and 32 in Fig. 2) for each oscillator output in order to drive the phase errors to zero. In the above code example, the update for each phase estimate may be
 35 computed by:

```

EPS*sign((a[i%M]-aa[i%M])-(b[i%N]-bb[i%N]))

```

This line of code can be viewed as corresponding to the scaled binary difference
 40 between the corrected RX and TX phase errors. The sign() function simply returns 1 or -1, depending on the sign of the phase error. This is used as follows to simulate a binary quantized phase error consistent with the algorithm of Table 1.


```

# implement 1st order BB control
# return only the sign of error
function sign(a) {
    if (a>0) {
5         return 1
    } else {
        return -1
    }
}
10

```

In order to simulate the situation in which the remote transmitter is not capable of being calibrated by information gathered at the receiver, the TX adaptation is inhibited by commenting out the line of code:

```

15 # aa[i%M] += EPS*sign((a[i%M]-aa[i%M]) -
    (b[i%N]-bb[i%N]))

```

Fig. 3A is an example embodiment of one possible implementation of the present invention in which a stand-alone receiver, RX 41, calibrates itself without providing any communication back to the remote transmitter, TX 42. As stated above, this implementation would be useful for a receiver that is required to be interoperable with industry-standard transmitter parts that may not be capable of calibration in accordance with the present invention, or in a simplex application where there is no mechanism for providing communication from RX 41 to TX 42. If the RX 41 has an oscillator with an odd number of phases, then all phases will eventually be used to sample edges of the incoming data from TX 42. Odd receiver timing events are used to sample transmitter transitions, which are determined by the transmitter timing events.

Fig. 3B shows an enhancement of the configuration shown in Fig. 3A in which periodic calibration data is transmitted from the local RX 43 back to itself, as indicated by arrow 44, and to remote TX 45, as indicated by arrow 46. Likewise, periodic calibration data is transmitted from remote RX 47 back to itself, as indicated by arrow 48, and to remote TX 49, as indicated by arrow 51. Such a system could be implemented for a proprietary full-duplex link, or for a full-duplex link in which a standard is adopted to provide for the flow of low bandwidth calibration data from an RX to a remote TX.

Fig. 3C shows a system embodiment in which a transceiver IC chip is architected to have a both transmitter, TX 52, and a receiver, RX 53. During power up, a switch 54 on the RX 53 input is switched to the calibration position connecting the RX 53 input to the TX 52 output. The RX 53 then generates calibration values for both the TX 52 and RX 53 oscillators. Once the calibration has converged, the values are frozen and stored in local memory (not shown). This approach has the advantage of correcting deterministic phase errors in both the RX 53 and the TX 52. In this system, it is still possible to actively calibrate the RX 53, even after power up, in the manner discussed above with reference to Figs. 3A and 3B.

In Fig. 3D, a system is shown in which the TX 55 has a duplicate receiver (calRX) 56 comprised of a third oscillator and sampling structure which is used to continuously calibrate the TX 55 and calRX 56. The receiver RX 57 calibrates itself. This system has good flexibility in that it requires no support in the communication channel for transmitting calibration information over a link, as is the case with the transceiver 52/receiver pair 53 shown in Fig. 3C. Furthermore, the system shown in Fig. 3D operates continuously (i.e., continuously calibrates), and is compatible with equipment installed at the remote end that does not use calibration in accordance with the present invention. Of course a possible disadvantage of the system of Fig. 3D is that it requires more hardware than the systems shown in Figs. 3A – 3C.

In the embodiments of Figs. 3B, 3C, and 3D, the RXs must somehow communicate the m calibration terms back to the TX phase shifter circuitry. To accomplish this, the RX preferably unambiguously labels the m indications so that the TX can apply the corrections to the proper phases. In accordance with the present invention, two general mechanisms are proposed for this purpose, although the present invention is not limited to these mechanisms. In systems that calibrate a local TX, such as in the systems demonstrated by Figs. 3C and 3D, once the RX is locked to the TX, it is possible for the RX circuitry to access the state of the TX phases directly. A phase-slip detector can be used to monitor the RX and TX phase alignment to ensure that no information is transferred until the clocks are stable.

In systems where a remote TX is calibrated, such as in the system demonstrated by Fig. 3B, the RX may derive the phase labeling from some characteristic of the data stream. For example, in the art of data coding, the manner in which a stationary, unique marker can be embedded in a data sequence is well known. This can be accomplished with, for example, training sequences, sync characters, and

master transitions. The TX can either send such markers aligned to a known phase, or, alternatively, send a marker in conjunction with a label that designates the phase of the marker. Upon receipt of such a marker, the RX would either set its local modulo m counter to the agreed upon phase or to the value given in the label. By this

5 mechanism, the RX is able to uniquely indicate to the TX the phase to which each correction signal applies.

It should be noted that the present invention has been described with reference to example embodiments, and that the present invention is not limited to the embodiments described herein. Those skilled in the art will understand, in view of the

10 discussion provided herein, that modifications can be made to the embodiments described above without deviating from the scope of the present invention. For example, although the circuit 10 shown in Fig. 2 has been described as comprising certain components, those skilled in the art will understand that different components can be used to implement such a circuit. Also, the circuit 10 can be embedded in a

15 single IC or implemented using discrete components. Also, the circuit 10 can be implemented solely in hardware (e.g., with gates in an IC), or as a combination of hardware and software, as will be understood by those skilled in the art. For example, the ATB Detector 30 may be implemented in software that performs the Alexander Truth Table algorithm, whereas the other components may be implemented in

20 hardware either as components with an IC or as discrete components.